Large Scale Cloud-Based Deformable Registration for Image Guided Therapy

Shahram Mohrehkesh[1], Andriy Fedorov[3], Arun Brahmavar Vishwanatha[2], Fotis Drakopoulos[2], Ron Kikinis[3], Nikos Chrisochoides[2]

[1] Department of Computer and Information Sciences, Temple University

[2] Center for Real-Time Computing, Old Dominion University

[3] Brigham and Women's Hospital, Harvard Medical School

shahram@temple.edu, nikos@cs.odu.edu

Abstract- We present a feasibility study using cloud resources for computing the deformable registration or non-rigid registration (NRR) of brain MR images for Image Guided Neurosurgery (IGNS). We consider the use of cloud resources in two scenarios. First, we describe a workflow implementation to enable speculative computation of registration to improve confidence in the result and assist in retrospective evaluation of the method. We evaluate the use of computing and storage capabilities of the cloud to handle more than 6 TB of images. Second, we evaluate the feasibility of large scale running NRR on the cloud to provide timely execution of the most time-consuming components of the registration in short duration of a brain surgery. Our preliminary results indicate that the cloud provides practical and cost-effective means to support IGNS. In addition, cloud resources could be used to improve the accuracy of NRR up to 57%.

Keywords- Image Registration, Cloud, Big Data, Large Scale, Speculative Execution, Image-Guided Neurosurgery.

I. INTRODUCTION

Image guided procedures become increasingly used in a range of clinical applications [1]. Non-rigid registration is one of the key enabling technologies in facilitating image-guided interventions. It enables the use of imaging data obtained from different imaging devices and across time-points, and facilitates integration of this imaging data into intra-procedural workflows. Image-guided neurosurgery has been one of the most actively investigated clinical research applications, where non-rigid registration widely improves accuracy of tumor resection and reduce patient morbidity by enabling the overlay of the pre-procedural structural and functional data over the intra-procedural imaging. A clinically practical non-rigid registration method should consider the following factors: speed, robustness, and accuracy. The registration should be done within a time period compatible with the clinical workflow constraints to provide timely responses to the surgeons. The registration results should not be susceptible to imaging artifacts. The registration results should also realistically reflect the physical deformation of the tissue. Recently, Liu et al. [2] developed ITK filters for physics-based non-rigid registration (PBNRR), which satisfy the following requirements: account for tissue properties in the registration; improve accuracy compared to rigid registration and reduce execution time [2].

Despite the improvements in accuracy of PBNRR compared to rigid registration, registration accuracy can still be improved. Previous studies [3] show that varying the values of registration input parameters may result in higher accuracy. However, the optimum values of these parameters are difficult to identify. For example, there is no consensus about the true values for the physical properties of the live tissue in the biomechanics community[4]. Likewise, depending on the properties of the intra-operative MRI (e.g., quality and modality), the best choice of similarity metric to be used for the registration block matching component may not be clear [5]. Registration accuracy may also be affected by the values of block matching input parameters, such as block size and window size. The optimum values for block size and window size may depend on the properties of the images and the scale of the brain shift [5].

In this work we approach the registration problem as a speculative computation process, where to improve the accuracy of registration, several PBNRR instances are run with various input parameters and similarity metrics to find the parameter setting with maximum registration accuracy. In the current protocol of NRR, there is usually significant time (≈ 30 min) between the acquisition of the first intra-operative scan and the MRI showing brain shift (Figure 1). This time can be used to perform intra-operative comparison of the similarity metrics. Evaluating similarity measures on a single Pentium IV computer takes about two months to finish [5]. Evidently, such computations can be done intra-operatively only with the aid of large scale computing.

We propose to utilize large scale computing resources to find the optimal values of input parameters to improve the accuracy of registration. The approach to address the problem of optimal parameter selection is to use speculative execution of the registration [6]. The idea behind speculative execution is to compute multiple registrations on the same input images using different parameter settings. Variability between results computed in this way allows us to estimate the sensitivity of the method by automatically calculating certain metrics to assure registration accuracy.

To enable speculative execution as well as a large-scale study of PBNRR, many storage and computing resources are required simultaneously. With recent advances in cloud computing, many researchers are evaluating the usage of cloud resources for various big data and computationally intensive applications [7] including medical image processing applications. In most studies, however, applications do not require real-time response as they do in IGNS. Therefore, they do not report the delay and scheduling issues when computational resources in the cloud are used. In this paper, we first study the use of storage and computational resources in the cloud to provide a computational platform for PBNRR during IGNS. Next, we use the results of speculative execution to improve our understanding of the registration performance using varying input parameters.

The remainder of this paper is organized as follows. In Section II, we review the role of cloud computing in medical image processing applications. In Section III, registration is described in detail. The idea of speculative execution of PBNRR is explained in Section IV. In Section V, we present the results of speculative execution of PBNRR both on a private cloud and on the Microsoft Azure public cloud. The paper concludes with future research directions outlined in Section VI.

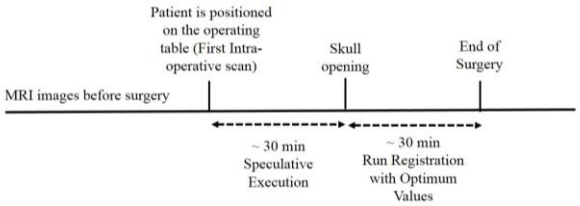

Fig. 1 Timeline of Brain Surgery

II. Cloud Computing and Medical Applications

Prior to cloud computing, we [3] and other researchers used grid computing to evaluate medical image applications. In grid computing, a lack of central resource management hinders the usability to run an application over a collection of loosely connected resources. Representative projects are MammoGrid [10] and the Biomedical Informatics Research Network (BIRN) [11]. By design, the Grid infrastructure developed either for general purpose Grid computing in our case TeraGrid [3] or by these projects were not general enough to support image processing applications like NRR. Several efforts have been made [12],[13] to use a general purpose research grid environment in the context of specific medical applications, but the absence of global resource management and the need to reserve resources limit the effectiveness of these grids. Moreover, in the case of intra-operative processing for IGNS, the actual time and length of the phase of the surgery where NRR will be critical is not precisely known in advance. The surgery may also be canceled or postponed due to last minute considerations such as changes in the patient's condition. Some research efforts [14]-[22] have designed solutions specifically for NRR on a grid, but network latency and issues in batch schedulers are such that they do not work during IGNS.

We envision that cloud computing [7] can provide a scalable and efficient solution for medical image processing applications, such as NRR during IGNS. Figure 2 illustrates the deployment of the cloud for NRR during IGNS. The pre-operative MRI images and intra-operative images are sent to and stored in the cloud. The NRR runs in speculative execution mode and the results of the registration with the highest accuracy are returned to the neurosurgeon. For subsequent registrations during the surgery, the parameters that provided the highest accuracy are used.

The use of cloud computing provides several advantages. First, resources in the cloud can be requested on demand; there is no need to reserve resources prior to the surgery. The demand can be adapted based on the circumstances, such as the number of patients or concurrent surgeries. Requesting resources on demand is very cost efficient for hospitals, as they do not pay for resources that they do not use. Nonetheless, the feasibility of using the cloud to optimize costs requires further investigation. This paper is a first step in this effort and focuses on real-time large scale accurate non-rigid registration.

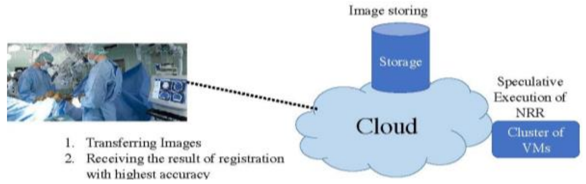

Fig. 2 Cloud Architecure for Running Speculative Execution of NRR

III. Non-rigid Registration of Brain MRI for IGNS

Image-guided neurosurgery aims for maximum tumor removal with minimal damage to surrounding healthy tissues. Pre-operatively acquired medical image data, such as MRI, identify tumor and critical brain regions with high precision. Yet, as the tumor resection progresses, brain deformation is unavoidable. Shift of the brain tissue invalidates pre-operative data and additional processing is required to account for the deformation. One of the available approaches to this problem is to acquire sparse images during IGNS, and update the pre-operative data according to the deformation observed in the intra-operative images. The specific image processing operation that aligns two images using high order transformation is known as non-rigid registration.

The details of the robust non-rigid registration method are given in [2,18]. Here, we summarize the main points of the approach. Registration consists of the pre-operatively initialization part and the intra-operative computation part. The goal of registration is to enable enhanced visualization of the brain under intra-operative deformation. Thus, the pre-operative (floating) image is deformed to match the intra-operative (reference) scan. An intra-operative image is acquired in the following cases: immediately after the patient's head is fixed for the surgery (first intra-operative scan), after the skull opening, at any time when the surgeon suspects significant shift of the brain, or when there is a need to verify residual tumor volume.

In pre-operative processing, multi-modal high-resolution scans are acquired prior to surgery to identify the tumor and critical regions in its vicinity, and to develop the appropriate resection strategy. The intra-cranial cavity is segmented, and the patient-specific biomechanical model is constructed for the subsequent application of Finite Element Method.

In intra-operative processing, rigid alignment of pre-operative data to the first intra-operative scan is the first processing task. This part of the intra-operative computation is not time-critical: there is typically sufficient delay between the first intra-operative scan and the skull opening. The time-critical component of the computation begins with the acquisition of a scan showing brain deformation. Initially, the locations of the floating image with the highest intensity variance in the surrounding region are selected to be registration points or landmarks. The sparse displacement field between the floating and reference images is estimated with the aid of volumetric block matching [17].

Depending on the properties of the intra-operative MRI (e.g., quality and modality), the choice of the best similarity metric to be used for block matching may not be clear [5]. At the same time, in the current protocol of NRR, there is usually significant time (approximately 30 min, as illustrated in Figure 1) between the acquisition of the first intra-operative scan and the MRI showing brain shift. The block matching result contains outlier displacements. The challenges in this phase include determining removal of outliers and the approximation of brain deformation from a sparse and irregular set of displacements. A mesh model of the intracranial cavity is used to approximate the brain shift using the finite element method (FEM).

We use Physics Based Non-Rigid Registration (PBNRR) for image registration. PBNRR framework is built in the National Library of Medicine Insight Segmentation and Registration Toolkit (ITK) [2]. The PBNRR module includes three components: (i) Feature Point Detection: identify small image blocks (landmarks) that have rich structural information in the pre-operative MRI; (ii) Block Matching: calculate displacement for each image block to generate a sparse deformation field; (iii) Robust Finite Element (FE) Solver: estimate entire brain deformation based on the sparse deformation field estimated above.

In PBNRR, we employ two image-to-mesh conversion methods for generating a mesh of the intra-cranial segmentation. The first method is Delaunay-based method [18], it recovers the isosurface of the biological object with geometric/topologic guarantees, and meshes the underlying volume with good shape tetrahedra. In this method, the parameter σ specifies the size of the Delaunay mesh, such that, the smaller the σ, the larger the mesh. The second method is Lattice Based Method (CBC3D) [19], it generates meshes of high geometric and topologic fidelity while it smoothens the mesh to provide a certain degree of visual reality within the requested fidelity The parameter *size* determines the lattice size and the parameter *fidelity* determines the desired mesh fidelity.

IV. Speculative Execution of PBNRR

Registration accuracy, robustness, and performance can all be affected by the values of NRR parameters. The optimal values of these parameters are difficult to identify. For example, there is no consensus about the true values for the physical properties of live tissue in the biomechanics community [4]. The optimal values for block size and window size used during block matching depend on the properties of the images and the scale of brain shift. The optimal similarity metric to be used during block matching is also not always known [5].

The practical problem of performing speculative execution is the enormous amount of computation required. For example, for sequential block matching with 100K registration points using Normalized Cross Correlation similarity metric, we need more than 11 hours of computation on a Intel Xeon 3.7GHz workstation. Considering that there may be 3-4 different similarity metrics that must be evaluated, a range of valid values for the outlier rejection in the solver, and a need to assess the accuracy of each NRR result, the total time required to perform an exhaustive evaluation of a single dataset would be days.

Cloud resources provide two important applications for PBNRR and IGNS: (1) intra-operative speculative execution, and (2) time-critical intra-operative computation of NRR with sub-optimal parameter settings. In the first application, PBNRR speculative execution finds the sub-optimal parameter settings, for the specific case, to improve accuracy. Next, in the time critical part of the surgery, the cloud resources are used to run PBNRR with the optimal parameter setting to achieve the result in real-time. To evaluate the performance of speculative execution for PBNRR, we used our local private on-premise cloud (Turing cluster) at Old Dominion University, as well as Microsoft Azure [7],[23]. Using these clusters, we analyzed more than 6TB of images.

A. On-premise Cluster

Turing cluster is mainly composed of Dell c8220 nodes, each with 128 GB memory and 16 cores. For our project, we had a quota of 160 cores. This means that at any point of time, we could run 10 instances of PBNRR simultaneously, where we chose to allocate 16 cores for each run of PBNRR. To parallelize and schedule job instances of PBNRR at Turing cluster, we used the array batch job submission mechanism of Sun Grid Engine (SGE) [24]. This mechanism allows to submit several jobs at the same time, and resources were assigned to jobs as soon as they were available.

B. Azure-based Cluster

Azure is Microsoft’s cloud solution, which provides several services to run various types of applications in the cloud [23]. We use Azure services to i) storage to manage the images, and ii) virtual machines (VMs) [25] to build a Windows High Performance Computing cluster.

We use the Windows HPC cluster [26] to run the speculative execution PBNRR. A Windows HPC cluster consists of a Head node, Proxy nodes and Computational nodes, as is illustrated in Figure 3. The Head node is an Azure VM that hosts a Windows Server and the HPC Manager application that manages the jobs submitted to the cluster. The Proxy nodes are Azure VM’s instances that balance job loads among computational nodes, as shown in Figure 3. The computational nodes are the VMs that run the job given to them by the Head node.

In the Windows HPC cluster, individual execution of an application- here PBNRR- is called a task, and a group of tasks- here, speculative execution of PBNRR with various parameters- is called a job.

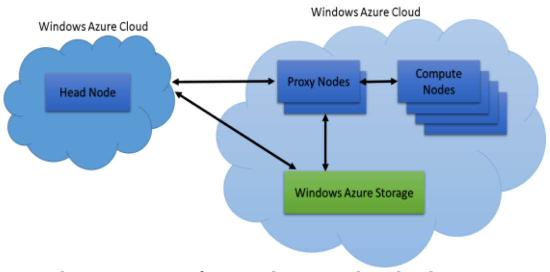

Fig. 3 Nodes in a Microsoft HPC cluster on the Cloud.

TABLE 2 PATIENT INFORMATION FROM EIGHT CASES

Case#	Tumor location	Histopathology
1	R occipital	Anaplastic Oligodendroglioma WHO III/IV
2	L posterior temporal	Glioblastoma WHO IV
3	R frontal	Oligodendroglioma WHO II/IV
4	R occipital	N/A
5	R frontal	Oligoastrocytoma WHO II/IV
6	L frontal	Oligodendroglioma WHO II/IV
7	R frontal	Oligodendroglioma WHO II/IV
8	R occipital	N/A

Our Microsoft Azure subscription granted us 32 cores and 10 TB of storage. We set up a cluster with a four cores Head node. The Head node VM location was set as EAST US. We used four cores for Proxy nodes. The remaining 24 cores were used to create virtual machines for computational nodes. We used A1 (1 core, 1.75GB RAM), A2 (2 core, 3.5GB RAM), A3 (4 core, 7GB RAM), A4 (8 core, 14GB RAM) VMs for computational nodes. This allowed us to have a cluster with 3, 6, 12, or 24 computational nodes.

V. RESULTS OF SPECULATIVE EXECUTION

In this section, we present our results of the speculative execution of PBNNR. We first describe the specifications of PBNRR experiments, and then present the improvement in the accuracy of registration by speculative execution. We also evaluate the overhead time of running jobs on the Azure cloud. Finally, we provide a cost comparison of running jobs on a private versus public cloud.

A. Experiments

We performed speculative execution of PBNNR by varying the values of several parameters (see Table 1) such as block size, window size and block connectivity. These variations produced approximately 10,000 jobs.

TABLE 1 VALUES FOR PARAMETERS IN SPECULATIVE EXECUTION.

Block Size (X=Y=Z)	3, 5, 7
Window Size (X,Y,Z)	5, 11, 15
Block Connectivity	vertex, edge, face
Rejection Steps	5, 10
Approximation Steps	2, 5, 10
Delaunay Mesh	σ = 2, 3, 4, 5
CBC3D Mesh	size = 6, 8, 10, and fidelity = 0.9

We conducted experiments on the registration between pre-operative MRI and intra-operative MRI. The image datasets came from public cases from the Surgical Planning Laboratory (SPL) at Harvard Medical School [27]. Table 2 lists patient information including tumor location, and histopathology.

B. Accuracy Measurements

As a measure of the registration accuracy, we used the one-directional Hausdorff Distance (HD) [22] which computes the alignment errors after a rigid and a non-rigid registration. The smaller the HD value, the better the alignment.

TABLE 3 THE REGISTRATION ERROR EVALUATED BY HD (IN MM) FOR 8 CASES. THE DEFAULT PARAMETERS FOR PBNRR FOR ALL CASES ARE: BLOCK RADIUS: [3,3,3], WINDOW RADIUS: [11,11,11], SELECTION FRACTION: 0.05, REJECTION FRACTION: 0.25, OUTLIER REJECTION STEPS: 10, APPROXIMATION STEPS: 10, YOUNG MODULUS: 694 PA, POISSON RATIO: 0.45.

Case#	**RR**	**PBNRR-default**	**PBNRR with speculative**	**PBNRR Improvement % (Default to Speculative)**
1	25.980	20.099	5.6569	71.855
2	9.110	4.690	2.2361	52.322
3	9.433	5.385	2.2361	58.475
4	9.695	7.000	2.2361	68.056
5	6.708	4.123	2.2361	45.765
6	8.062	3.605	2.2361	37.972
7	11.575	7.000	2.8284	59.594
8	14.352	9.949	3.0000	69.846
Average				57.986

In Table 3, we show the registration accuracy of the PBNRR, for the image cases when default parameters are used, versus speculative execution, with approximately 10,000 various input parameter combinations. These experiments produced more than 6 TB of images. As shown, we found a lower error value in registration for a combination of input parameters in speculative execution as compared with default parameters. The accuracy improvement varies between 37% to 71% for the 8 cases. The average improvement is 57.986%. The interesting observation is that, aside from image case #1, we achieved the error of less than 3 mm for all image scenarios, which is very close to the least error possible. Indeed, the minimum amount of error in registration when there is no brain resection, and deformation is only due to the movement of CF inside the brain, is between 0.75-1 mm. We could not achieve a better accuracy for image case #1 due to high distortion in the original images, i.e., 25.98 mm in rigid registration.

We found that block size and the size of mesh has the highest impact in the error. For most images the higher the value of block size and mesh size, the lower the error. Inspired by these results, we evaluated the results for conditions where only block size, window size, size of mesh, and type of meshes are varied. This approach results in 63 various combinations of input parameters, which is way less than our initial 10,000 combinations, and also a reasonable number of jobs to be run in the time constraint scenario of neurosurgeries. The minimum value for image registration for these 63 combinations is shown in Table 4.

TABLE 4 THE MINIMUM REGISTRATION ERROR- EVALUATED BY HD.

Case#	Number of Combinations		Difference (%)	Delaunay Mesh	CBC3D Meshes
	10, 000	63			
1	5.6569	5.6569	0.00	5.6569	6.4031
2	2.2361	2.2361	0.00	2.2361	2.4495
3	2.2361	2.2361	0.00	2.4494	2.2361
4	2.2361	3.0000	34.16	3.6055	3
5	2.2361	2.4495	9.54	2.4495	2.4495
6	2.2361	2.2361	0.00	2.2361	2.4495
7	2.8284	3.000	6.07	3	3
8	3.0000	4.5826	52.75	6.4031	4.5826
Average			12.8		

As can be observed, in four image cases, the results of image registration with the 63 combinations are the same as the scenario with 10,000 combinations. The average difference of results is only about 12%. The 30 minutes time constraint allows us to run about 300 instances of PBNRR; therefore, we believe that a smart search on the various input parameters combinations can produce the same exact results as the full 10,000 combination scenario. Developing such a smart search space methodology is part of our future work.

C. Speculative Execution in the Azure Cloud

In Microsoft Azure cluster, in addition to speculative execution of PBNRR to improve the accuracy in image registration, we are interested in evaluating the overhead of running jobs in a HPC cluster built in the cloud. For this purpose, we vary the number of computational nodes and the number of tasks (while varying the values of registration input parameters) when executing the speculative execution of PBNRR.

To compute the overhead time, we collect two metrics for each PBNRR run as a task on the Azure cloud. PBNRR run time (T_1): This time is recorded by the PBNRR, which is the execution time of the PBNRR application. Cloud run time (T_2): The time since the task is submitted from the Head node to a Computational node until the status of the task is complete at the Head node.

The Microsoft HPC cluster stores T_1 and T_2 in a database, which is connected to the Head node. The overhead time, then, is defined as $T_2 - T_1$. This overhead time includes queuing time to submit the jobs and time to collect the results from the Computational nodes to the Head node. Figure 4 illustrates the average overhead time for various numbers of tasks and various numbers of Computational nodes. The average overhead time is less than 3.5 seconds in most scenarios, which is not a significant overhead when compared with the average 60-300 seconds runtime for each instance of PBNRR itself. Therefore, PBNRR can run on the cloud-based cluster for IGNS without any significant overhead time.

D. Cost Comparison

In this section, we compare the costs associated with running speculative execution on the public cloud- here, Microsoft Azure- versus a similar private on-premise cluster. The goal is to evaluate whether it is cost efficient for a hospital to run speculative execution of PBNRR on the cloud. We evaluate a scenario where Computational nodes have 8 cores, 14 GB RAM, and 2 TB hard disk. A machine with these specifications costs approximately $2309. This type of machine costs $0.64/hr (~$476/mo) in Microsoft Azure. We consider a cluster of 30 Computational nodes for speculative execution of PBNRR. With this number of computational nodes, we can ensure that enough PBNRR instances will run to improve the accuracy of registration during a surgery.

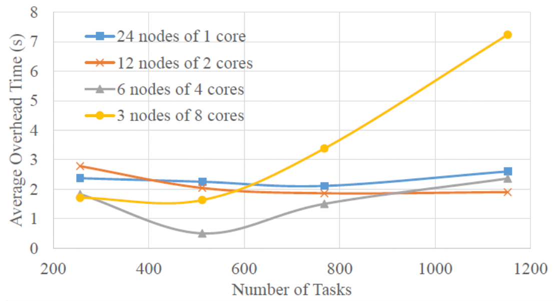

Fig. 4 Average Overhead Time for Various Number of Tasks

For the public cloud, we assume that the computational nodes are required for only 10 hours each day, since surgery occurs during limited hours of the day. For a private cluster, in addition to the cost of Computational nodes, other costs such as labor, network switches, software, electricity, space, and maintenance should be included. In our calculations, we considered $5000 for labor and a low estimate of $1000 for other associated costs each month. The duration of return on investment (ROI) for a private cluster is usually considered to be 3 to 5 years [28].

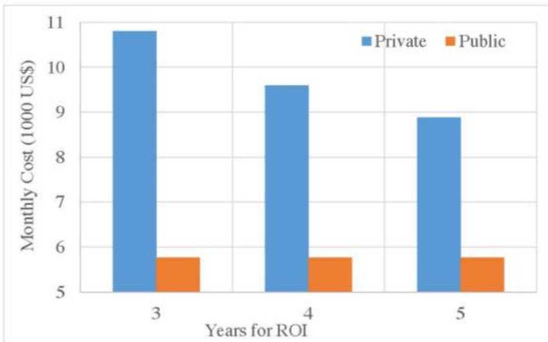

Fig. 5 Cost Comparison of Private vs. Public cloud.

Figure 5 illustrates the monthly cost for a privately owned cluster versus a cluster built in the Microsoft Azure cloud. Even in five years ROI, a public cloud would cost 2/3 of a private cloud. Therefore, running the speculative execution on the cloud is a cost effective solution for a hospital. Moreover, the cost per month is less than $6000. If 30 cases are performed each month, each would cost approximately $200, which is low in comparison to the total cost of surgery. For this additional cost, the quality of IGNS could be significantly improved, as presented in Section V.

VI. CONCLUSIONS AND FUTURE WORK

We have evaluated the feasibility of implementing speculative execution of the PBNRR method on a private and a public cloud. Our work shows that it is possible to use a cluster of nodes on the cloud to perform large scale data and compute intensive image processing applications. In particular, we showed improvement in the accuracy of registration in average by 53%. We ran several instances of PBNRR with various input parameters to find the combination that resulted in minimal error. The overhead of running speculative execution of PBNRR on a public cloud is negligible, and this is a cost efficient approach for a hospital that wants to implement IGNS solutions.

In the future, we plan to expand our experiments in several ways. We are interested in evaluating PBNRR performance with more image cases. Moreover, a brute-force speculative execution would not be very useful because as input parameters are varied, the number of combinations grow exponentially. Therefore, a smarter search space algorithm is required to have a scalable solution based on the speculative execution results. The approach we are investigating is to find a subset of input parameters that has a higher probability of producing the lowest errors and run the speculative execution only for those potential candidates.

Acknowledgments

The authors acknowledge funds from NSF grants CCF-1139864 and CCF-1439079 and the Richard T. Cheng Endowment and John Simon Guggenheim Memorial Foundation We would also like to acknowledge Dennis Gannon from Microsoft for access to Azure resources for this research. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NSF and private foundations.

References

[1] Ferenc A. Jolesz, "Intraoperative Imaging and Image-Guided Therapy", SpringerLink, (2014).

[2] Y. Liu, A. Kot, F. Drakopoulos, C. Yao, A. Fedorov, A. Enquobahrie, O. Clatz, and N. P. Chrisochoides, "An ITK implementation of a physics-based non-rigid registration method for brain deformation in image-guided neurosurgery," Frontiers in Neuroinformatics 8(33) (2014).

[3] A. Fedorov, B. Clifford, S. Warfield, R. Kikinis, and N. Chrisochoides, "Non-rigid registration for image-guided neurosurgery on the teragrid: A case study," tech. rep., tech. report WM-CS-2009-05, Dept. of Computer Science (2009).

[4] H. Delingette and N. Ayache, Soft tissue modeling for surgery simulation, vol. XII of Hand book of Numerical Analysis: Special volume: Computational models for the human body, 453–550. Elsevier, Netherlands, 1 ed. (2004).

[5] D. Skerl, B. Likar, and F. Pernuˇs, "A protocol for evaluation of similarity measures for nonrigid registration," Medical Image Analysis 12(1), 42–54 (2008).

[6] F. Ino, Y. Kawasaki, T. Tashiro, Y. Nakajima, Y. Sato, S. Tamura, and K. Hagihara, "A parallel implementation of 2-d/3-d image registration for computer-assisted surgery," Int J for Bioinformatics Research and Applications 2(4), 341–357 (2006).

[7] D. Gannon, D. Fay, D. Green, K. Takeda, and W. Yi, "Science in the cloud: Lessons from three years of research projects on microsoft azure," in Proceedings of the 5th ACM Workshop on Scientific Cloud Computing, ScienceCloud '14, 1–8, ACM, (New York, NY, USA) (2014).

[8] P. M. Mell and T. Grance, "Sp 800-145. The NIST definition of cloud computing," tech. rep., Gaithersburg, MD, United States (2011).

[9] I. Foster, "What is the grid? a three point checklist," white paper, (2002), Available at http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf.

[10] "MammoGrid Project," (2008). http://www.cems.uwe.ac.uk/cccs/project.php?name=mammogrid.

[11] "Biomedical Informatics Research Network," (2008). http://www.nbirn.net.

[12] S. Dong, J. Insley, N. T. Karonis, M. E. Papka, J. Binns, and G. Karniadakis, "Simulating and visualizing the human arterial system on the TeraGrid," Future Generation Computer Systems, 22, 1011–1017 (2006).

[13] S. Manos, S. Zasada, M. D. Mazzeo, R. Haines, G. Doctors, S. Brew, R. Pinning, J. Brooke, and P. V. Coveney, "Patient specific whole cerebral blood flow simulation: A future role in surgical treatment for neurovascular pathologies," in Proc. of Teragrid'08, (2008).

[14] N. Archip, O. Clatz, S. Whalen, D. Kacher, A. Fedorov, A. Kot, N. Chrisochoides, F. Jolesz,A. Golby, P. M. Black, and S. K. Warfield, "Non-rigid alignment of pre-operative MRI, fMRI, and DT-MRI with intra-operative MRI for enhanced visualization and navigation in image-guided neurosurgery," Neuroimage, 35, 609–624 (2007).

[15] N. Chrisochoides, A. Fedorov, A. Kot, N. Archip, O. Clatz, R. Kikinis, and S. K. Warfield, "Toward real-time image guided neurosurgery using distributed and grid computing," in Proc. of the ACM/IEEE Conference on Supercomputing, (2006).

[16] A. Majumdar, A. Birnbaum, D. J. Choi, A. Trivedi, S. K. Warfield, K. Baldridge, and P. Krysl, "A dynamic data driven grid system for intra-operative image guided neurosurgery," in Proc. of ICCS 2005, 672–679 (2005).

[17] O. Clatz, H. Delingette, I. F. Talos, A. J. Golby, R. Kikinis, F. A. Jolesz, N. Ayache, and S. K. Warfield, "Robust non-rigid registration to capture brain shift from intra-operative MRI," IEEE Tran. on Medical Imaging 24(11), 1417–1427 (2005).

[18] P. Foteinos, N. Chrisochoides, "High quality real-time Image-to-Mesh conversion for finite element simulations," J. of Parallel and Distributed Computing, 74 (2), 2123-2140 (2014).

[19] Y. Liu, P. Foteinos, A. Chernikov, N. Chrisochoides, "Mesh deformation-based multi-tissue mesh generation for brain images", J. of Engineering with Computers, 29(4), 305-318 (2012).

[20] F. Drakopoulos, P. Foteinos, Y. Liu, and N. P. Chrisochoides, "Toward a real time multi-tissue adaptive physics-based non-rigid registration framework for brain tumor resection," Frontiers in Neuroinformatics 8(11) (2014).

[21] M. Vangel, A. Fedorov, W. W. III, and C. Tempany, "Statistical framework for characterization of deformable registration performance," Tech. report, Department of Radiology, Brigham and Women's Hospital, Harvard Medical School, Boston, MA, USA, (2012).

[22] F. Commandeur, J. Velut, and O. Acosta, "A VTK algorithm for the computation of the hausdorff distance," VTK J. (2011).

[23] Microsoft, "Microsoft azure for research overview," (2013). Available at http://research.microsoft.com/en-us/projects/azure/windows-azure-for-research-overview.pdf

[24] "Grid engine manual pages," (2014). Available at http://gridscheduler.sourceforge.net/htmlman/manuals.html

[25] Microsoft, "Getting started with Microsoft azure virtual machines," (2013). Available at http://research.microsoft.com/en-us/projects/azure/getting-started-with-vms-in-windows-azure.pdf

[26] Microsoft, "Microsoft HPC pack (windows HPC server)," (2014). Available at https://technet.microsoft.com/en-us/library/cc514029.aspx

[27] I.-F. Talos and N. Archip, "Volumetric non-rigid registration for MRI-guided brain tumor surgery," Tech. Report, Surgical Planning Laboratory, Department of Radiology, Brigham and Women's Hospital, Harvard Medical School, Boston, MA, USA (2007).

[28] A. Kraut, S. Moretti, M. Robinson-Rechavi, H. Stockinger, and D. Flanders, "Phylogenetic code in the cloud–can it meet the expectations?," Studies in health technology and informatics 159, 55–63 (2010).